

WHAT IS CLAIMED IS:

1 1. A method of managing a generational memory, the method comprising:
2 sampling, at run-time of an execution sequence, lifetimes of a representative
3 subset of memory objects in the generational memory; and
4 pretenuring, based on the sampled lifetimes, at least some of the memory
5 objects allocated from the generational memory during the execution
6 sequence.

1 2. The method of claim 1,
2 wherein the pretenuring is performed for those of the memory objects for
3 which corresponding sampled lifetimes exceed a first metric.

1 3. The method of claim 2,
2 wherein the pretenuring is reversed, if corresponding sampled lifetimes fall
3 below a second metric.

1 4. The method of claim 1,
2 wherein the pretenuring is performed while corresponding sampled lifetimes
3 exceed a metric.

1 5. The method of claim 1,
2 wherein the sampling includes sampling of representative subsets for plural
3 categories of the memory objects; and
4 further comprising:
5 allocating the memory objects using category-specific allocation
6 functionality; and
7 selectively modifying the category-specific allocation functionality to
8 pretenure, on subsequent allocations, memory objects
9 corresponding to those of the categories for which the sampled
10 lifetimes exceed a metric.

1 6. The method of claim 5,

wherein the selective modification of the category-specific allocation functionality includes instantiating category-specific allocation methods that allocate new objects of a corresponding category directly into a tenured generation of the generational memory.

7. The method of claim 1, wherein the sampled lifetimes are characterized as per-category, mean lifetimes.

8. The method of claim 1, wherein the sampled lifetimes are characterized as per-category, distributions of lifetimes.

9. The method of claim 5, wherein the categories are object class-specific.

10. The method of claim 5, wherein the categories are call-site specific.

11. The method of claim 5, wherein the categories correspond to activation record stack profiles.

12. The method of claim 5, wherein the categories correspond to both type of memory object and call-site for allocation thereof; and wherein the selective modification of the category-specific allocation functionality includes in-lining instructions at the corresponding call-site, the in-lined instructions allocating new memory objects of the corresponding type directly into a tenured generation of the generational memory.

13. The method of claim 1, wherein the sampling includes sampling of representative subsets of the memory objects for plural categories thereof; and

wherein the pretenuring is performed on a category-specific basis.

14. The method of claim 1, further comprising:
 establishing weak references to respective of the sampled memory objects and
 associating allocation-time information therewith; and
 identifying those of the sampled memory objects that become unreachable
 using the weak references.

15. The method of claim 14, wherein the weak references include one of:
 phantom references;
 references of strength less than any other reference by which an unreachable
 one of the sampled objects may become reachable; and
 references of strength less than any other reference employed in the
 computational system.

16. The method of claim 1, further comprising:
 selecting the representative subset based on allocation buffer overflow.

17. The method of claim 1, further comprising:
 selecting the representative subset using per class allocator functionality.

18. The method of claim 1, further comprising:
 selecting the representative subset based on identity of an allocating thread.

19. The method of claim 1, further comprising:
 coincident with allocation of memory objects of the representative subset,
 establishing weak references thereto and associating therewith
 information indicative of at least allocation time and call site.

20. A method of operating an automatically reclaimed storage environment in
 accordance with object lifetime statistics, the method comprising:
 selecting representative subsets of memory objects for each of plural
 categories thereof;

5 sampling, during a program execution, lifetimes of memory objects from the
 6 representative subsets; and
 7 tailoring, during the program execution, a storage management action based
 8 on the sampled lifetimes for a corresponding one of the categories.

1 21. The method of claim 20,
 2 wherein the storage management action includes pretenuring subsequently
 3 allocated memory objects of the corresponding category.

1 22. The method of claim 20,
 2 wherein the storage management action includes promoting memory objects
 3 of the corresponding category to a particular generation.

1 23. The method of claim 20,
 2 wherein the storage management action includes steering, on promotion,
 3 memory objects of the corresponding category to a particular store.

1 24. The method of claim 20,
 2 wherein the storage management action includes allocating subsequently
 3 allocated memory objects of the corresponding category from a
 4 particular store.

1 25. The method of claim 24,
 2 wherein the particular store is selected from the set of thread-local storage and
 3 global storage, based on the sampled lifetimes.

1 26. The method of claim 20,
 2 wherein the storage management action includes modifying a category-
 3 specific allocator in accordance with a storage management policy
 4 adapted to the sampled lifetimes; and
 5 instantiating the modified, category-specific allocator.

1 27. The method of claim 20,

2 wherein the automatically reclaimed storage environment includes a
 3 generational garbage collector; and
 4 wherein the storage management action includes generation selection for
 5 unsampled instances of the memory objects.

1 28. A storage management facility for a computational system, the storage
 2 management facility comprising:
 3 an object sampler operable to sample lifetimes of at least a subset of objects
 4 instantiated in the computational system during execution of a
 5 program; and
 6 a storage allocation facility operable during the execution of the program to
 7 allocate new objects corresponding to respective of the sampled
 8 objects based at least in part on the sampled object lifetimes.

1 29. The storage management facility of claim 28,
 2 wherein the object sampler samples lifetimes on a per object category basis;
 3 and
 4 wherein operation of the storage allocation facility is particular to each object
 5 category and based at least in part on the lifetimes of the sampled
 6 objects corresponding thereto.

1 30. The storage management facility of claim 28,
 2 wherein the storage allocation facility includes category-specific allocators;
 3 and
 4 wherein, in response to respective of the sampled object lifetimes exceeding a
 5 metric, the category-specific allocators are modified at run-time to
 6 pretenure objects allocated thereby.

1 31. The storage management facility of claim 28, wherein the categories
 2 correspond to one or more of:
 3 object type;
 4 allocation call site;
 5 activation record stack state;

6 thread id; and
7 receiver object.

1 32. The storage management facility of claim 28,
2 wherein the object sampler is responsive to transition of at least one of the
3 sampled objects from a reachable state to an unreachable state.

1 33. The storage management facility of claim 28,
2 wherein the object sampler employs a weak reference construct of the
3 computational system to identify those of the sampled objects that have
4 become unreachable.

1 34. The storage management facility of claim 33,
2 wherein the weak reference construct includes a phantom reference.

1 35. The storage management facility of claim 33,
2 wherein the weak reference construct includes a virtual-machine-level weak
3 reference of strength less than any other weak reference by which an
4 unreachable instance of the sampled objects may become reachable.

1 36. The storage management facility of claim 33,
2 wherein the weak reference construct includes a virtual-machine-level weak
3 reference of strength less than any other weak reference employed in
4 the computational system.

1 37. The storage management facility of claim 28, further comprising:
2 a garbage collector,
3 wherein the object sampler is responsive to a storage management event of the
4 garbage collector affecting one or more of the sampled objects.

1 38. The storage management facility of claim 37,
2 wherein the storage management event includes collection of one or more of
3 the sampled objects.

1 39. The storage management facility of claim 37,
2 wherein the storage management event includes promotion of one or more of
3 the sampled objects from a younger generation to an older generation.

1 40. The storage management facility of claim 37, embodied as a computer
2 program product.

1 41. A computer program product encoded in at least one computer readable
2 medium, the computer program product comprising:
3 at least one functional sequence for maintaining per-category object lifetime
4 statistics based on a sampled subset of objects using weak references
5 and associated allocation time information; and
6 at least one functional sequence for tenuring objects in accordance with those
7 of the object lifetime statistics corresponding thereto.

1 42. A computer program product as recited in 41,
2 wherein the tenuring sequence is instantiated for a particular category when
3 corresponding object lifetime statistics exceed a metric.

1 43. A computer program product as recited in 42,
2 wherein the tenuring sequence replaces a category-specific allocator.

1 44. A computer program product as recited in 42,
2 wherein the tenuring sequence is replaced by a non-tenuring, category-specific
3 allocator when corresponding object lifetime statistics fall below a
4 second metric.

1 45. A computer program product as recited in 41,
2 wherein the at least one computer readable medium is selected from the set of
3 a disk, tape or other magnetic, optical, or electronic storage medium
4 and a network, wireline, wireless or other communications medium.

1 46. An apparatus comprising:

2 means for sampling instances of software objects to maintain lifetime
3 predictions for categories thereof; and
4 means for altering object category-specific storage management policies at
5 run-time in response to the lifetime predictions.

1 47. The apparatus of claim 46, further comprising:
2 means for reversing the object category-specific storage management policies
3 at run-time in response to the lifetime predictions.